REMARKS

Claims 1-5 and 7-20 are pending in the above-captioned patent application. Claims 1, 10, 16 and 19 are independent claims.

Applicant amended independent claim 1 to clarify that the branching operation is performed if a state, of a state name specified in the branch instruction is a specified value, with the state indicating the availability of a resource of the data processing apparatus.

Applicant amended independent claims 10, 16 and 19 to clarify those claims. Applicant also amended the preambles of claims 1 and 19 to clarify that it is a branch instruction stored on the computer readable medium that causes the operations recited in independent claims to occur. In addition, applicant added new claims 21 and 22, which depend from independent claim 1, to include the feature that the resource corresponding to the state name includes a queue, and the feature that in some embodiments the queue is a FIFO request receive queue. Support for these features is provided, for example, at page 14, lines 14-19 of the originally filed application.

The examiner rejected claims 1 and 19 under 35 U.S.C. §101. Specifically, the examiner stated:

> Although applicant amended the program product residing on a computer readable medium for causing the execution, no practical application can be found in the clamed invention. The examiner understands the program product being stored in the computer readable medium to cause execution , but the focus is not on whether the steps taken to achieve a particular result is useful, tangible, concrete, but rather the final result is useful, tangible and concrete (page 20 of the internal guideline newly updated on 01/17/06). Claim 1 recites to cause execution, and it is read as a step taken to achieve practical result, but it is not a final result. Claim 1 also recites to branch to an instruction at a specific address at a value of availability of resource, but no substantial practical application can be found in the claim. As such, the claim is not useful, tangible and concrete, and is therefore non-statutory. (Office Action, page 3, Paragraph 4).

Applicant contends that there is no support for the examiner's position regarding the requirement that the final result needs to be useful. MPEP §2106.IV.B.2.(b)(ii) (relating to computer-related processes), provides:

> **A claim is limited to a practical application when the method, as claimed, produces a concrete, tangible and useful result; i.e., the method recites a step or act of producing something that is concrete, tangible and useful. See AT&T, 172 F.3d at 1358, 50 USPQ2d at 1452.**

...

Examples of this type of claimed statutory process include the following:

- A computerized method of optimally controlling transfer, storage and retrieval of data between cache and hard disk storage devices such that the most frequently used data is readily available.

- A method of controlling parallel processors to accomplish multi-tasking of several computing tasks to maximize computing efficiency. See, e.g., In re Bernhart, 417 F.2d 1395, 1400, 163 USPQ 611,616 (CCPA 1969).

Clearly then, subject matter that relates for improving or maximizing computer efficiency is patentable subject matter. Applicant's claim 1 is directed to computer efficiency in that the branch instruction, when executed, expedites performance of the branch instruction in that the condition for branching (namely, the state of a state name, which corresponds to the availability of a resource) is already specified in the branch instruction.

Applicant contends that it is unclear what criteria the examiner is using to define what is a "final result." Moreover, it is unclear upon what authority the examiner relies on to support his conclusion that there is a requirement that: "the final result is useful, tangible and concrete" for finding statutory subject matter. The examiner makes reference to page 20 of an internal guideline newly updated on 01/17/06." This reliance is improper since the notice apparently was not published and thus was not noticed for public comment. More importantly however, this purported notice and the examiner's conclusion are in direct contravention of established Board of Patent Appeals and Interferences and Federal Circuit precedents. See for example. *In re Lowry*, 32 F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994), where the court specifically found that:

> The Board reversed the 35 U.S.C. Section 101 rejection. The Board found that claims 1 through 5, directed to a memory containing stored information, as a whole, recited an article of manufacture. The Board concluded that the invention claimed in claims 1 through 5 was statutory subject matter. Lowry 32 F.3d at

In *Lowry*, the claims were found statutory by the Board because the claims were to a memory, i.e., an article of manufacture. The Board acknowledged the statutory nature of the memory, but then proceeded to apply the so-called printed matter doctrine and was subsequently

reversed by the Federal Circuit. Applicant's claim 1 recites an article of manufacture, namely "a computer program product residing on a computer readable medium." The Patent Office's own published guidelines for patenting of computer related inventions does not support the position taken by the examiner. See also *In re Warmerdam*, 33 F.3d 1354, 31 U.S.P.Q.2d 1754 (Fed Cir. 1994) in which the court held that when functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized. Further, at page 1759, a claim to computer having a specific data structure stored in memory was held statutory product-by-process claim, whereas claims to a data structure per se were held non-statutory.

Moreover, in any event, applicant contends that completion of the operations caused by the execution of the branch instruction is the final result, and that result is concrete, tangible, and useful.

The applicant further notes that the language in the preamble of amended claims 1 and 19 conforms to the conventional form widely used to recite computer program product claims. Such claims are regularly used and appear in numerous issued patents including, for example, recently issued U.S. Patent Nos. 6,961,787, 6,961,686, and 6,954,833.

Applicant thus traverses the examiner's rejection of claims 1 and 19 under 35 U.S.C. §101.

The examiner rejected claims 1 and 19 under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 4,454,595 to Cage.

Additionally, the examiner maintained his rejection of claims 1, 10 and 19 under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 6,275,508 to Aggarwal. The examiner also maintained his rejections of claims 1-20 under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 5,724,563 to Hasegawa, and further maintained his rejections of claim 1 under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 5,640,538 to Dyer.

Applicant's independent claim 1 recites the feature to "cause an executing instruction stream to branch to an instruction at a specified address if a state, of a state name specified in the branch instruction is a specified value, the state indicating the availability of a resource of the data processing apparatus." Thus, a decision to branch to a specified address depends on the

availability of a resource of the data processing apparatus, as identified by a state name that is specified in the branch instruction being executed. Applicant's branch instruction facilitates performance of both a comparison operation and a branching operation.

In contrast, none of the references cited by the examiner discloses or suggests a branching instruction that causes a branching operation to be performed based on the availability of a resource specified in the branch instruction.

Cage describes a buffer for use with a word processing disk controller. The buffer is connected to a communication bus, which in turn is connected to a direct-memory-access controller (Abstract). Cage's apparatus also includes a data path sequencer 244 that controls DMA operations (col. 10, lines 49-55). In relation to the disk write/DMA operations performed by Cage's apparatus, Cage explains:

> **If all the data bytes required for this operation have not been transferred, the sequencer forces a branch from step 524 to step 528, to check the status of buffer space available. If buffer space is still available, the sequencer branches to step 526 again to determine whether a serializer request is pending. If buffer space is not available, however, the sequencer generates a DMA request to fetch the next data byte, step 530. The sequencer advances from step 530 to step 532, where it checks DMA READY to determine whether the floppy disk processor 116 has presented the next data byte on the bus. If the next data byte has not been presented, the sequencer checks buffer available status again, step 528. If no buffer space is still available, the sequencer then attempts to load the first 256-byte data block into the sequencer RAM data buffer 260 before it initiates transfer to the disk. The sequencer remains in this loop until the floppy disk processor 116 has presented a data byte or until 500 milliseconds have elapsed, at which point the operation aborts. (Col. 14, lines 1-20)**

Thus, while the data path sequencer 244 performs, among other things, an operation to check the availability of buffer space, and can subsequently perform other operations based on that determination, the sequencer does not perform branching operations that cause an executing instruction stream to be directed to a different address in the instruction stream. The use of the term "branching" in the above excerpt from Cage, and elsewhere, refers to jumping to a different points in a flowchart (e.g., the flowchart shown in Cage's FIG. 5A). At no point, however, does Cage describe a branch instruction, and certainly does not describe a branch instruction in which the state of a state name specified in the branch instruction is determined. Thus, Cage does not disclose or suggest at least the feature to: "cause an executing instruction stream to branch to an

instruction at a specified address if a state, of a state name specified in the branch instruction is a specified value, the state indicating the availability of a resource of the data processing apparatus," as required by applicant's independent claim 1.

As for Aggarwal, as explained in applicant's November 8, 2005, Amendment in Reply to Action of August 16, 2005, (hereinafter the "previous Amendment in Reply"), Aggarwal describes a system for processing datagram headers during traverses from one interface of a networking device to another (Abstract). Aggarwal's system includes a sequencer unit that controls the selection of data from input data stream (FIGS. 1 and 12, and Abstract). The sequencer unit is controlled by a word instruction called the Write Control Store (WCS) (see col. 3, lines 61-63). As described in Aggarwal, the WCS includes various fields, shown in FIG. 12, amongst which is a conditional branch field. Aggarwal explains that the condition branch tells the sequencer unit "to jump to one of the multiple addresses in its address field, each one representing the next possible address, if the current instruction is a Conditional Branch ..." (col. 10, lines 24-27). No additional explanations regarding Aggarwal's conditional branch are provided.

Thus, although Aggarwal mentions a branch instruction, Aggarwal does not disclose that a state name, which is to be evaluated to determine the availability of a resource, is specified in the branch instruction. Accordingly, Aggarwal neither discloses nor suggests at least the feature of "cause an executing instruction stream to branch to an instruction at a specified address if a state, of a state name specified in the branch instruction is a specified value, the state indicating the availability of a resource of the data processing apparatus," as required by independent claim 1.

Hasegawa discloses a pipeline processor that can execute predictive branch instructions (Abstract). Hasegawa further describes that the format of its predictive branch instructions includes a region 21 that stores an opcode, a region 22 for specifying a branch target address, and a region 23 for storing the number of at least one instruction which is to be executed in succession after the predictive branch instruction is given before the control flow is changed (FIG. 2 and col. 6, lines 1-6). For example, as illustrated in FIGS. 5A and 5B, the value 3 stored in region 23 of the predictive branch instruction helps control when the branching operation

would be performed after execution of the branch instruction. Hasegawa's predictive branch instruction, however, does not include a region for specifying a state name.

Furthermore, as explained in applicant's previous Amendment in Reply, Hasegawa's predictive branching is performed based on the execution result 109 produced by executing an instruction on execution section 11 of Hasegawa's pipeline processor (FIG. 9, col. 11). Specifically, the execution of an instruction on execution section 11 (the executed instruction is not the predictive branch instruction) causes a number of flags on the condition code 61 of the judging section 13 to be set in accordance with the result outcome produced by the execution of the instruction (FIG. 9 and col. 11, lines 20-32). These flags include the Zero flag Z, the negative flag N, the Carryover flag C, and the overflow flag V.

Hasegawa further explains that after the execution result 109 is produced, the opcode in the region 21 of the predictive branch instruction is input from the instruction decoding section 3 to the branch condition judging section 62 (col. 11, lines 33-35). For example, if the opcode received from the predictive branching instruction is "100", branching will occur if the Z flag has been set as a result of the execution of a preceding instruction (see Table 1 at col. 11). Thus, Hasegawa's branching decisions are based on the value of the Z, N, C and V flags of the condition code 61 unit, and not on the state of a state name specified by the branching instruction, where the state is indicative of the availability of a resource.

Accordingly, Hasegawa neither discloses nor suggests at least the feature of "cause an executing instruction stream to branch to an instruction at a specified address if a state, of a state name specified in the branch instruction is a specified value, the state indicating the availability of a resource of the data processing apparatus," as required by independent claim 1.

Dyer describes a programmable timing mark sequencer that automatically analyzes a sequence of data bits on a data input line (Abstract). Dyer's programmable timing mark sequencer uses an instruction whose format is described in FIG. 8. As shown in Table 1, describing the fields of the instruction shown in FIG. 8, the instruction includes a Branch Address field which specifies the branch address that is taken if the branch condition specified in the Branch Type field is true (col. 11, lines 6-8). As Dyer further describes, if the Branch Type field is set to 1, the branching operation is unconditional, whereas if the Branch Type field is set to 0 then branching will be performed on the presence of bit HRBIT.

The examiner stated:

> Dyer taught a conditional branch on a bit value of high resolution bit to drive the sequence of timing mark (see col.11, lines 6-8). The timing mark was a resource. Dyer also taught branch based on a state name (branch type), the state indicated the available resource (1,0). Whether branch was conditional or unconditional , it branched if the state was one of the 1,0 branch type values. (Office Action, page 9, paragraph 30).

Applicant disagrees with the examiner's characterization of the HRBIT as a resource.

The high resolution bit input line is used to drive the timing mark sequencer. Specifically, Dyer explains that "[i]f the appropriate timing pattern is detected in the sequence of high resolution data bits HRBIT, timing mark sequencer 240 drives servo timing mark signal STM active" (col. 6, lines 45-48). Thus, the HRBIT is an external time-dependent signal that causes, if the Branch Type bit of instruction 800 is set to 0, to branch to the address specified in the Branch Address field. The HRBIT, therefore, is not a resource of the data processing apparatus (e.g., a resource such as a queue). Thus, Dyer does not disclose or suggest at least the feature of "cause an executing instruction stream to branch to an instruction at a specified address if a state, of a state name specified in the branch instruction is a specified value, the state indicating the availability of a resource of the data processing apparatus," as required by independent claim 1.

Because none of the references cited by the Examiner discloses or suggests, alone or in combination, at least the feature of "cause an executing instruction stream to branch to an instruction at a specified address if a state, of a state name specified in the branch instruction is a specified value, the state indicating the availability of a resource of the data processing apparatus," independent claim 1 is therefore patentable over the cited prior art.

Claims 2-5, 7-9 and 21-22 depend from independent claim 1 and are therefore patentable for at least the same reasons as independent claim 1.

Independent claims 10, 16 and 19 recite the feature of "evaluating a value of a state name specified in a branch instruction, the value of the state name indicating the availability of a resource of the processor; and performing a branching operation based on the value of the specified state name being set or cleared," or similar language. At least this feature is not disclosed by the prior art cited by the examiner for reasons similar to those provided with respect

to independent claim 1. Accordingly, Independent claims 10, 16, and 19 are patentable over the prior art.

Claims 11-15 depend from independent claim 10 and are therefore patentable for at least the same reasons as independent claim 10. Claims 17-18 depend from independent claim 16 and are therefore patentable for at least the same reasons as independent claim 16. Claim 20 depends from independent claim 19 and is therefore patentable for at least the same reasons as independent claim 19.

Additionally, the examiner rejected claim 9 under U.S.C. §102(b) as being anticipated by U.S. Patent No. 5,724,563 to Hasegawa. Specifically, the examiner argued that:

> As to claims 8, 9 Hasegawa also included performing the branch based on specified name (see branch on over flow set and branch on overflow clear in Table 1, see the flags set and reset in col.l 1 [sic] lines 14-35, see also the encoded flags in col.1, lines 42-52). As to the parallel processor, see the pipeline processor in col. 5, lines 30-31. (Office Action, Page 6, Paragraph 18)

Claim 9, which depends from independent claim 1, recites the feature "wherein the state name is the name assigned to an executing context." Thus, branching to a specified address takes place, for example, based on whether or not a specified context is executing (basing the branching decision on the availability of the data processor apparatus' processing capabilities, or resources).

As noted above, Hasegawa conditional branches are based on the values of a number of flags that reflect the outcome that resulted from the execution of an instruction. Hasegawa, therefore, does not base its branching decisions on the state of a specified state name indicating the availability of a resource of the data processing apparatus, and certainly does not base branching decision on whether or not a particular named context is executing, as required by applicant's claim 9. Accordingly, claim 9 is patentable over the cited art.

It is believed that all the rejections and/or objections raised by the examiner have been addressed.

In view of the foregoing, applicant respectfully submits that the application is in condition for allowance and such action is respectfully requested at the Examiner's earliest convenience.

All of the dependent claims are patentable for at least the reasons for which the claims on which they depend are patentable.

Canceled claims, if any, have been canceled without prejudice or disclaimer.

Any circumstance in which the applicant has (a) addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner, (b) made arguments for the patentability of some claims does not mean that there are not other good reasons for patentability of those claims and other claims, or (c) amended or canceled a claim does not mean that the applicant concedes any of the examiner's positions with respect to that claim or other claims.

Please apply the required fee of $50 for excess claim fees, and any other charges or credits to deposit account 06-1050, referencing attorney docket 10559-306US1.

Respectfully submitted,

Date: April 27, 2006

\Ido Rabinovitch\
Ido Rabinovitch
Attorney for Intel Corporation
Reg. No. L0080

Fish & Richardson P.C.
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21318043.doc